Dr. Kari Jakbboor, Ph.D  
Curriculum Developer,  
American College of Technology,  
Missouri, USA.  

Er. Chandramohan, M.S  
System Specialist - OGP  
ABB Australia Pvt. Ltd., Australia.  

Dr. S.K. Singh  
Chief Scientist  
Advanced Materials Technology Department  
Institute of Minerals & Materials Technology  
Bhubaneswar, India  

PROF. Dr. Sharath Babu, LL.M Ph.D  
Dean, Faculty Of Law  
Karnatak University Dharwad,  
Karnataka, India  

Dr. SM Kadri, MBBS, MPH/ICHD,  
FFP Fellow, Public Health Foundation of India  
Epidemiologist Division of Epidemiology and Public Health  
Kashmir, India  

Dr. Bhumika Talwar, BDS  
Research Officer  
State Institute of Health & Family Welfare  
Jaipur, India  

Dr. Tej Pratap Mall Ph.D  
Head, Postgraduate Department of Botany  
Kisan P.G. College, Bahraich, India.  

Dr. Arup Kanti Konar, Ph.D  
Associate Professor of Economics  
Achhruram, Memorial College  
SKB University, Jhalda, Purulia,  
West Bengal, India  

Dr. S. Raja Ph.D  
Research Associate,  
Madras Research Center of CMFR ,  
Indian Council of Agricultural Research  
Chennai, India  

Dr. Vijay Pithadia, Ph.D,  
Director - Sri Aurobindo Institute of Management  
Rajkot, India.  

Er. R. Bhuvanewari Devi M.Tech, MCIHT  
Highway Engineer, Infrastructure,  
Ramboll, Abu Dhabi, UAE  

Sanda Maican, Ph.D.  
Senior Researcher,  
Department of Ecology, Taxonomy and Nature Conservation  
Institute of Biology of the Romanian Academy,  
Bucharest, ROMANIA  

Dr. Damarla Bala Venkata Ramana  
Senior Scientist  
Central Research Institute for Dryland Agriculture (CRIDA)  
Hyderabad, A.P, India  

PROF. Dr. S.V. Kshirsagar, M.B.B.S, M.S  
Head - Department of Anatomy,  
Bidar Institute of Medical Sciences,  
Karnataka, India.  

DR ASIFA NAZIR, M.B.B.S, MD  
Assistant Professor Dept of Microbiology  
Government Medical College, Srinagar, India.  

Dr. Amita Puri, Ph.D  
Officiating Principal  
Army Inst. Of Education  
New Delhi, India  

Dr. Shobana Nelasco Ph.D  
Associate Professor,  
Fellow of Indian Council of Social Science Research (On Deputation) ,  
Department of Economics,  
Bharathidasan University, Trichirappalli, India  

M. Suresh Kumar, PHD  
Assistant Manager,  
Godrej Security Solution,  
India.  

Dr. T. Chandrasekaraya, Ph.D  
Assistant Professor,  
Dept Of Population Studies & Social Work,  
S.V.University, Tirupati, India.
ITEM-BASED COLLABORATIVE FILTERING RECOMMENDER SYSTEM

PROF. M.S. PRASAD BABU*
BODDU RAJA SARATH KUMAR**
MENTEY SRIRAJ (M.TECH) ***

*Dept. of CS & SE, AUCE, Andhra University, Andhra Pradesh, India
** Professor, Dept. of CSE, Lenora College of Engineering, Andhra Pradesh, India
*** Dept. of CS & SE, AUCE, Andhra University, Andhra Pradesh, India

ABSTRACT:
Recommendation algorithms are best known for their use on e-commerce Web sites, where they use input about a customer’s interests to generate a list of recommended items. Many applications use only the items that customers purchase and explicitly rate to represent their interests, but they can also use other attributes, including items viewed, demographic data, subject interests, and favourite artists. Collaborative filtering is one of the most important technologies in electronic commerce. With the development of recommender systems, the magnitudes of users and items grow rapidly, resulted in the extreme sparsity of user rating data set. Traditional similarity measure methods work poor in this situation, make the quality of recommendation system decreased dramatically. Poor quality is one major challenge in collaborative filtering recommender systems. Sparsity of users’ ratings is the major reason causing the poor quality. To address these issues we have explored item-based collaborative filtering techniques. Item based techniques first analyze the user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users. This approach predicts item ratings that users have not rated, and then uses Pearson correlation similarity measurement to find the target items’ neighbors, lastly produces the recommendations. The experiments are made on a common data set using different recommender algorithms. The results show that the proposed approach can improve the accuracy of the collaborative filtering recommender system.

KEYWORDS: Recommender Systems, Collaborative Filtering, Sparsity, Item Based Collaborative Filtering
1. INTRODUCTION

People have used the phrase collective intelligence for decades, and it has become increasingly popular and more important with the advent of new communications technologies. Although the expression may bring to mind ideas of group consciousness or supernatural phenomena, when technologists use this phrase they usually mean the combining of behaviour, preferences, or ideas of a group of people to create novel insights. Collective intelligence was, of course, possible before the Internet. One of the most basic forms of this is a survey or census. Collecting answers from a large group of people draws statistical conclusions about the group that no individual member would have known by themselves. Building new conclusions from independent contributors is really what collective intelligence is all about.

A well-known example is financial markets, where a price is not set by one individual or by a coordinated effort, but by the trading behaviour of many independent peoples all acting in what they believe is their own best interest. Although it seems counterintuitive at first, futures markets, in which many participants trade contracts based on their beliefs about future prices, are considered to be better at predicting prices than experts who independently make projections. This is because these markets combine the knowledge, experience, and insight of thousands of people to create a projection rather than relying on a single person’s perspective.

Although methods for collective intelligence existed before the Internet, the ability to collect information from thousands or even millions of people on the Web has opened up many new possibilities. At all times, people are using the Internet for making purchases, doing research, seeking out entertainment, and building their own web sites. All of this behaviour can be monitored and used to derive information without ever having to interrupt the user’s intentions by asking him questions. There are a huge number of ways this information can be processed and interpreted. Here are a couple of key examples that show the contrasting approaches:
A. Wikipedia is an online encyclopaedia created entirely from user contributions. Any page can be created or edited by anyone, and there are a small number of administrators who monitor repeated abuses. Wikipedia has more entries than any other encyclopaedia, and despite some manipulation by malicious users, it is generally believed to be accurate on most subjects. This is an example of collective intelligence because each article is maintained by a large group of people and the result is an encyclopaedia far larger than any single coordinated group has been able to create. The Wikipedia software does not do anything particularly intelligent with user contributions—it simply tracks the changes and displays the latest version.

B. Google, mentioned earlier, is the world’s most popular Internet search engine, and was the first search engine to rate web pages based on how many other pages link to them. This method of rating takes information about what thousands of people have said about a particular web page and uses that information to rank the results in a search. This is a very different example of collective intelligence. Where Wikipedia explicitly invites users of the site to contribute, Google extracts the important information from what web-content creators do on their own sites and uses it to generate scores for its users.

2. Collaborative Filtering Based Recommender Systems

The term collaborative filtering was first used by David Goldberg at Xerox PARC in 1992 in a paper called “Using collaborative filtering to weave an information tapestry.”[3,9] He designed a system called Tapestry that allowed people to annotate documents as either interesting or uninteresting and used this information to filter documents for other people. There are now hundreds of web sites that employ some sort of collaborative filtering algorithm for movies, music, books, dating, shopping, other web sites, podcasts, articles and even jokes.

Online shopping site Amazon tracks the purchasing habits of all its shoppers, and when they log onto the site, it uses this information to suggest products shoppers might like. Amazon can even suggest movies shoppers might like, even if they have only bought books from it before. Some online concert ticket agencies will look at the history of shows viewers have seen before and alert them to upcoming shows that might be of interest. Sites like reddit.com let users vote on links to other web sites and then use user’s votes to suggest other links each individual might find interesting.
A collaborative filtering algorithm usually works by searching a large group of people and finding a smaller set with tastes similar to an individual. It looks at other things they like and combines them to create a ranked list of suggestions. There are several different ways of deciding which people are similar and combining their choices to make a list.

2.1 Challenges of User-based Collaborative Filtering Algorithms

User-based collaborative filtering systems have been very successful in past, but their widespread use has revealed some potential challenges such as:

**Sparsity:** In practice, many commercial recommender systems are used to evaluate large item sets (e.g., Amazon.com recommends books and CDnow.com recommends music albums). In these systems, even active users may have purchased well under 1% of the items (1% of 2 million books is 20,000 books). Accordingly, a recommender system based on nearest neighbor algorithms may be unable to make any item recommendations for a particular user. As a result the accuracy of recommendations may be poor.

**Scalability:** Nearest neighbor algorithms require computation that grows with both the number of users and the number of items. With millions of users and items, a typical web-based recommender system running existing algorithms will suffer serious scalability problems.

3. Item-Based Filtering

In cases with very large datasets, item-based collaborative filtering can give better results, and it allows many of the calculations to be performed in advance so that a user needing recommendations can get them more quickly.

The general technique is to pre compute the most similar items for each item. Then, when there is a need to make recommendations to a user, just look at his top-rated items and create a weighted list of the items most similar to those. The important difference here is that, although the first step requires us to examine all the data, comparisons between items will not change as often as comparisons between users. This means we do not have to continuously calculate each item’s most similar items—we can do it at low-traffic times or on a computer separate from your main application.
3.1 Item-based Collaborative Filtering Algorithm

In this section we study a class of item-based recommendation algorithms for producing predictions to users. Unlike the user-based collaborative filtering algorithm the item-based approach looks into the set of items the target user has rated and computes how similar they are to the target item \( i \) and then selects \( k \) most similar items \( \{i_1, i_2, \ldots, i_k\} \). At the same time their corresponding similarities \( \{s_{i1}, s_{i2}, \ldots, s_{ik}\} \) are also computed. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user’s ratings on these similar items. The process is shown in fig 1.

3.2 Finding Similar Items

The first step in computing the similarity of two items \( i_p \) and \( i_q \) (column vectors in the data matrix \( M \)) is to identify all the users who have rated (or visited) both items. Many measures can be used to compute the similarity between items. Here we present one such method.

3.3 Pearson Correlation Score

Sophisticated way to determine the similarity between items is to use a Pearson correlation coefficient. The similarity between two items \( i \) and \( j \) can be computed by
Here $U(i)$ includes all users who have rated on item $i$. Formally, $U(i) = \{u | ru,i \neq 0\}$. $U(i) \cap U(j)$ includes the users who have rated on both item $i$ and $j$. $u\_r$ is the average of user $u$'s ratings.

3.4 Prediction Computation

The most important step in a collaborative filtering system is to generate the output interface in terms of prediction. Once we isolate the set of most similar items based on the similarity measures, the next step is to look into the target users ratings and use a technique to obtain predictions.

3.5 Weighted Sum

To compute the recommendations or predictions to the user using the item similarity dictionary without going through the whole dataset. As the name implies, this method computes the prediction on an item $i$ for a user $u$ by computing the sum of the ratings given by the user on the items similar to $i$. Each ratings is weighted by the corresponding similarity $s_{ij}$ between items $i$ and $j$. we can denote the prediction $Pu,i$ as

$$Pu,i = \frac{\sum_{j \in \text{items similar to } i} s_{ij} \cdot Pu,j}{\sum_{j \in \text{items similar to } i} s_{ij}}$$
Table 1 Item recommendation for a user U

Shows the process of finding recommendations using the item-based approach.

Each row has a movie that User U has already seen, along with his personal rating for it. For every movie that User u haven’t seen, there’s a column that shows how similar it is to the movies U has seen—for example, the similarity score between Superman and The Night Listener is 0.103. The columns starting with R.x show my rating of the movie multiplied by the similarity—since user u rated Superman 4.0, the value next to Night in the Superman row is $4.0 \times 0.103 = 0.412$. The total row shows the total of the similarity scores and the total of the R.x columns for each movie. To predict what user u rating would be for each movie, just divide the total for the R.x column by the total for the similarity column. My predicted rating for The Night Listener is thus $1.378/0.433 = 3.183$.

4. MovieLens Dataset

MovieLens is a web-based research recommender system that debuted in Fall 1997.[5] Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 43000 users who have expressed opinions on 3500+ different movies. MovieLens was developed by the GroupLens project at the University of Minnesota which be can download the dataset from http://www.grouplens.org/node/12

![Table 1](image-url)
The archive contains several files, but the ones of interest are *u.item*, which contains a list of movie IDs and titles, and *u.data*, which contains actual ratings in this format:

```
196   242    3    881250949
186   302    3    891717742
 22    377    1    878887116
 244    51    2    880606923
 166    346    1    886397596
 298    474    4    884182806
```

Each line has a user ID, a movie ID, the rating given to the movie by the user, and a timestamp. Movie titles can be obtained, but the user data is anonymous, hence I worked with user IDs. The set contains ratings of 1,682 movies by 943 users, each of whom rated at least 20 movies.

### 5. Methodology

All our experiments were implemented using *Java Platform Standard Edition 6*. We ran all our experiments on a windows based PC with Intel Core 2 duo processor with processing speed of 2.00 GHz and 2GB of RAM.

### 6. Experimental Results

Below are the result pasted from the output of item based collaborative filtering algorithm

The result shown in screenshot1 is the profile of a user with the user id 3.

```
+-----------------+   USER PROFILE +-----------------+
|-----------------|-----------------|
User id           : 3
Age               : 23
Gender            : M
Users Occupation  : writer
user 3 has rated 54 Movies
+-----------------+   USER PROFILE +-----------------+
```
Screen Shot :1

Below screen shot is the extraction of movies from the total movies rated by the user 3

| Return of the Jedi (1983) | Action Adventure Romance Sci-Fi War |
| Devil’s Own, The (1997) | Action Drama Thriller War |
| Contact (1997) | Drama Sci-Fi |
| Event Horizon (1997) | Action Mystery Sci-Fi Thriller |
| Mimic (1997) | Sci-Fi Thriller |
| Chasing Amy (1997) | Drama Romance |
| Starship Troopers (1997) | Action Adventure Sci-Fi War |
| Good Will Hunting (1997) | Drama |
| Scream (1996) | Horror Thriller |

Screen shot: 2

Below screen shot is the extraction of movies from total predictions to user 3

| Back to the Future (1985) | Comedy |
| Faster Pussycat! Kill! Kill! (1965) | Action Comedy Drama |
| Hot Shots! Part Deux (1993) | Action Comedy War |
| Snow White and the Seven Dwarfs (1937) | Animation Children Musical |
| Right Stuff, The (1983) | Drama |
| Turbo: A Power Rangers Movie (1997) | Action Adventure Children |

Screen shot:3

6.1 Experiment Evaluation:

The most commonly used recommender metric for measuring the quality of recommendations is Mean Absolute Error (MAE). MAE is a measure of the deviation of recommendations from their true user-specified values.

The MAE is computed by first summing these absolute errors of the $N$ corresponding ratings-prediction pairs and then computing the average. The formula to calculate MAE is

$$ MAE = \frac{\sum_{i=1}^{N} |a_i - b_i|}{N} $$
The accuracy of the recommendations is calculated with MAE. The low value of MAE indicates the accuracy of recommendations.

We used MAE as our choice of evaluation metric to report prediction experiments because it is most commonly used and easiest to interpret directly.

The MAE values are calculated and tabulated in table 2. The influence of various nearest neighbours set on predictive validity is tested by gradually increasing the number of neighbours. Here it is observed that when Nearest Neighbour Set value increases the corresponding MEA decreased.

<table>
<thead>
<tr>
<th>Neighbour Set Size</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>0.95</td>
<td>0.93</td>
<td>0.89</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Table 2: Nearest neighbour set and MEA on predictive validity

Graphical representation of Neighbourhood size vs. MAE value is shown below

Graph 1: Graphical representation of Neighborhood size vs MAE value
7. CONCLUSION

In this paper, item based collaborative filtering algorithm has been implemented and evaluated. The quality of the predictions is evaluated with similar algorithms. The item based collaborative filtering algorithm overcomes the overheads that are associated with the used based collaborative filtering. Item-based filtering is significantly faster than user-based when getting a list of recommendations for a large dataset, but it does have the additional overhead of maintaining the item similarity table. Also accuracy difference depends on how it “sparse” the dataset. Item based collaborative filtering approach shows that it exhibits a behaviour that is equivalent to that of the best algorithms. The main aim of this paper is to improve the quality of the results and make it easier for the user to find relevant information from the internet.

References


